**Contents:**

**Introduction---------------------------------------------------------------------------------------------------------------------- 3**

**Solution of given topic (Comments)--------------------------------------------------------------------------------------- 3**

**Result (suggestions)----------------------------------------------------------------------------------------------------------- 16**

**References ---------------------------------------------------------------------------------------------------------------------- 16**

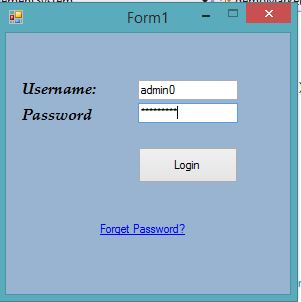
**Introduction:**

In modern life every industry should be digitalized. In retail sector it is essential topic. As it is known that supermarkets should handle huge amount of orders, sales and manage all data related to internal business processes. For this reason, several technical solutions are worked by companies. One of the common problems is to keep all information and details about products, orders , sales and etc. Instead of paper stuff, automated electron systems makes jobs done faster and smoothly. In this project 2 modules will be presented.

**Solution of given topic (Comments)**

Supermarket Management System is a huge enterprise system. Generally, it consists of several modules. In this project only admin and data entry modules are presented. Program has been written in C# Winforms. It is a desktop application. It runs on MySQL database. There are 9 forms for realizing given task.

**Login Page:**

****

Login Page is a first page of program. It checks users credentials and if it is a valid user it redirects user to a page according to his role. There are 2 user pages: Admin, DataMan.

*private void btnLogin\_Click(object sender, EventArgs e)// validating user*

*{*

*if (!IsFieldsEmpty(textboxUsername, textboxPassword))*

*{*

*//*

*setUserCredentials(textboxUsername.Text, textboxPassword.Text);*

*if (isUserFound(userName, userPassword))*

*{*

*setUserID();*

*openUsersPage(userRole);*

*}*

*}*

*}*

private bool IsFieldsEmpty(TextBox obj,TextBox obj2)

{

if(String.IsNullOrEmpty(obj.Text)|| String.IsNullOrEmpty(obj2.Text))

{

MessageBox.Show("Please fill all fields");

return true;

}

return false;

}

private bool isUserFound(string username, string password)

{

if(connectToDatabase())

{

try

{

string command = String.Format("select u\_role from storemanagement.users where u\_user\_name = '{0}' and u\_password = '{1}'", username, password);

MySqlCommand cmd = new MySqlCommand(command, conn);

MySqlDataReader reader = cmd.ExecuteReader();

if(reader.HasRows)

{

while (reader.Read())

{

userRole = reader.GetString(0);

}

bool ans = closeConnectionToDatabase();

// MessageBox.Show("YES");

return true;

}

else

{

MessageBox.Show("Access Denied. Username or password is wrong");

bool ans = closeConnectionToDatabase();

return false;

}

}

catch (Exception e) { MessageBox.Show(e.ToString()); bool ans = closeConnectionToDatabase(); };

}

// MessageBox.Show("Access Denied. Username or password is wrong");

return false;

}

private void setUserID()

{

string userid = DatabaseFunctions.selectScalar(

String.Format("select u\_user\_id from storemanagement.users" +

" where u\_user\_name = '{0}' and u\_password = '{1}';",userName,userPassword));

this.userID = userid;

}

private void openUsersPage(string userRole)

{

switch (userRole)

{

case "admin":

AdminPage admin = new AdminPage(userID);

// this.Hide();

admin.ShowDialog();

break;

case "dataman":

DatamanPage dp = new DatamanPage(userID);

dp.ShowDialog();

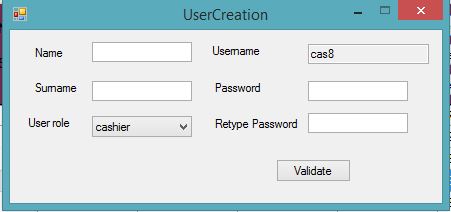
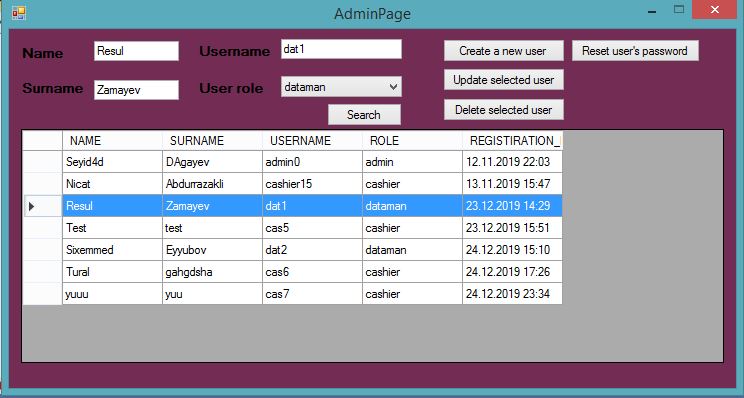
break;

default: break;

}

}

**Admin Page:**

It is an admin control panel where admin can create a user, list all users, search users filtering by name, surname, role, username, update users’ information, delete a user and reset their passwords to default password. When “Create a new user” button is clicked a new page opens and gets credentials of new user.

private void btnSearch\_Click(object sender, EventArgs e)

{

setAll();

bool ans = select(generateSelectQuery());

setAllToNull();

}

private string generateSelectQuery()

{

string query = "select u\_name as NAME,u\_surname as SURNAME,"

+ " u\_user\_name as USERNAME, u\_role as ROLE, u\_reg\_date as REGISTIRATION\_DATE"

+ " from " + databaseName + " where u\_status = 1 ";

if (name.Length > 0)

query += String.Format("and u\_name ='{0}' ", name);

if (surname.Length > 0)

query += String.Format("and u\_surname ='{0}' ", surname);

if (username.Length > 0)

query += String.Format("and u\_user\_name ='{0}' ", username);

if (userrole.Length > 0)

query += String.Format("and u\_role ='{0}' ", userrole);

return query;

}

private void add()

{

string query = "insert into " + databaseName + "(u\_user\_name, u\_name, u\_surname, u\_role,u\_password, u\_reg\_date, u\_status) values "

+ String.Format("('{0}','{1}','{2}','{3}','{4}',sysdate() ,1)",username,name,surname,userrole,password);

bool res = false;

//textBoxName.Text = query;

bool connect = connectToDatabase();

try

{

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = query;

cmd.Connection = conn;

cmd.ExecuteNonQuery();

textBoxName.Text = query;

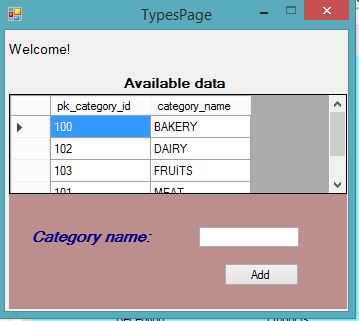
res = true;

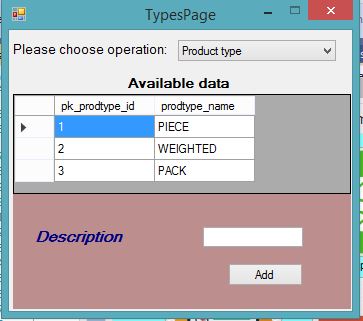
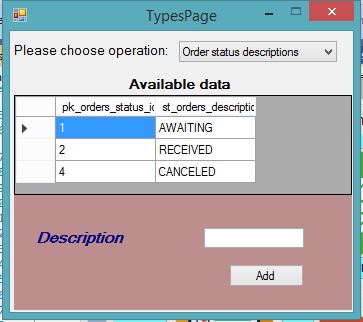
}

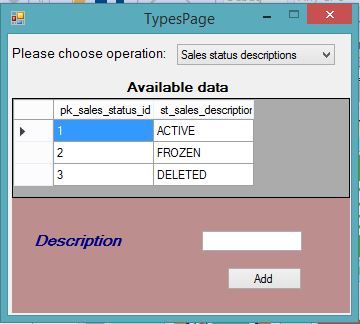
catch (Exception e) { MessageBox.Show("ADD function error:" + e.ToString()); }

bool close = closeConnectionToDatabase();

}

****

****

****

**DataMan Page:**

When “Categories” or “Types” is clicked, “Types Page” opens. “Categories” is for adding new category and listing all categories. Types opens a page where a combobox offers operations. After choosing operation related information is loading to datagridview.

private void Btn\_add\_Click(object sender, EventArgs e)

{

switch (currentWorkType)

{

case "categories":

insertTable("categories", "category\_name");

categoriesLoad();

break;

case "general":

switch (comboBox1.SelectedIndex)

{

case 0: //sales status

insertTable("status\_sales", "st\_sales\_description");

break;

case 1: //product

insertTable("product\_type", "prodtype\_name");

break;

case 2: //order

insertTable("status\_orders", "st\_orders\_description");

break;

default: break;

}

check();

break;

default: break;

}

}

private void insertTable(string tableName,string columnName)

{

string value = textBoxEntry.Text.Trim().ToUpper();

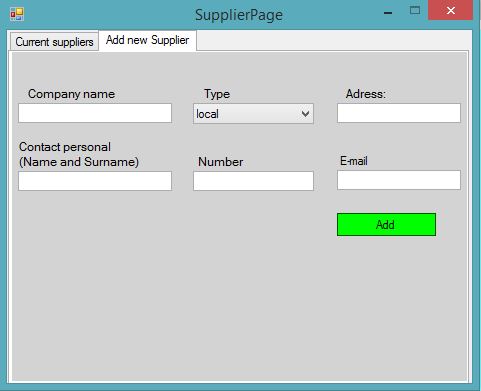
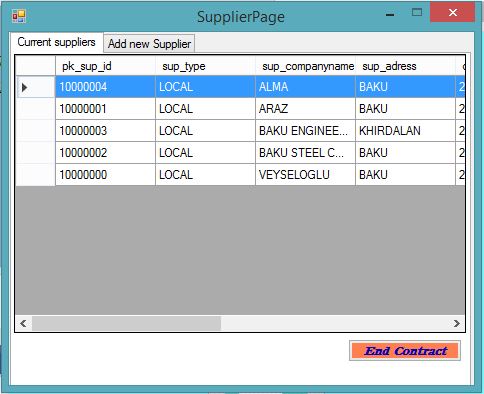
string query = String.Format("insert into storemanagement.{0}({1}) values('{2}');",tableName,columnName,value);

bool added = DatabaseFunctions.runner(query);

if (added) { MessageBox.Show("Succesfully added"); }

else { MessageBox.Show("Error in new creation in "+tableName); }

}



**Supplier Page:**

There is “Tab pane” with 2 tab controls “Current supplier” and “Add new Supplier”. Current supplier tab shows all suppliers and their information. When “End Contract” is clicked it sets “Contract end date” to the date of click. Add new Supplier pane holds combobox, textbox and button for inserting new data.

private void BtnEndContract\_Click(object sender, EventArgs e)

{

int count = 0;

for(int i = 0;i< dataGridView1.SelectedRows.Count;i++)

{

string id = dataGridView1.SelectedRows[i].Cells[0].Value.ToString();

string ended = dataGridView1.SelectedRows[i].Cells[8].Value.ToString().Trim();

string query = String.Format("update storemanagement.suppliers " +

"set end\_date = date(sysdate()) where pk\_sup\_id = {0}", id);

if (ended.Length == 0)

{

bool res = DatabaseFunctions.runner(query);

if(res == false) { count++; }

}

}

if(count > 0)

{ MessageBox.Show("Some items may not be ended"); }

else { MessageBox.Show("Successfully ended contracts"); }

showAllSuppliers();

}

private void showAllSuppliers() {

string query = "select \* from storemanagement.suppliers order by sup\_companyname asc;";

dataGridView1.DataSource = DatabaseFunctions.select(query);

}

private void BtnAdd\_Click(object sender, EventArgs e)

{

String companyName, type, adress, personNameSurname, number, email;

companyName = textBoxCompanyName.Text.Trim().ToUpper();

type = comboBoxType.SelectedItem.ToString().Trim().ToUpper();

adress = textBoxAdress.Text.Trim().ToUpper();

personNameSurname = textBoxNameSurname.Text.Trim().ToUpper();

number = textBoxNumber.Text.Trim();

email = textBoxEmail.Text.Trim().ToUpper();

if (String.IsNullOrEmpty(companyName) || String.IsNullOrEmpty(type) ||

String.IsNullOrEmpty(adress) || String.IsNullOrEmpty(personNameSurname) ||

String.IsNullOrEmpty(number) || String.IsNullOrEmpty(email))

MessageBox.Show("Please fill all fields!");

else {

string query = String.Format("insert into storemanagement.suppliers " +

"(sup\_type,sup\_companyname,sup\_adress,sup\_personalname," +

"sup\_telnumber,sup\_email,contract\_date) values('{0}', '{1}'," +

" '{2}', '{3}', '{4}', '{5}', sysdate());",

type,companyName,adress,personNameSurname,number,email);

bool check = DatabaseFunctions.runner(query);

if (check) { MessageBox.Show("Successfully added"); }

else { MessageBox.Show("Error in creating new supplier"); }

textBoxAdress.Text = null;

textBoxCompanyName.Text = null;

textBoxEmail.Text = null;

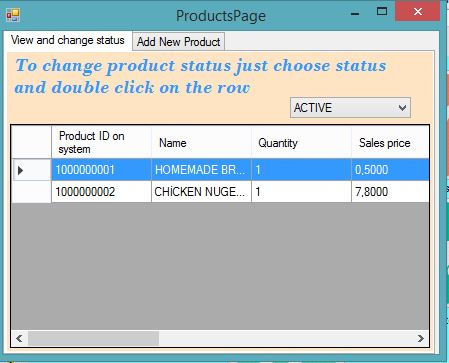
comboBoxType.SelectedIndex = -1;

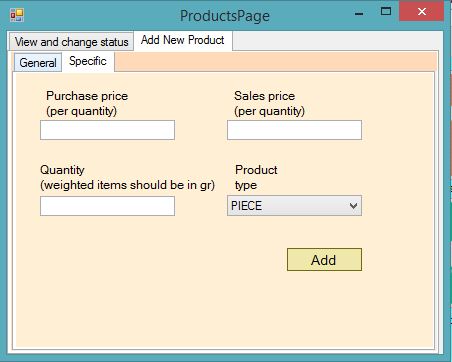
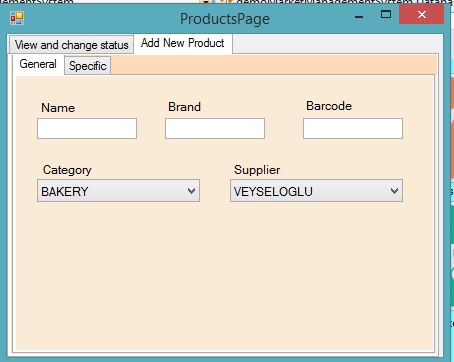
textBoxNameSurname.Text = null;

textBoxNumber.Text = null;

}

}





**Products page:**

There are 2 TabPane in Products page. First is for viewing data and changing status. Combobox gets all status types from database. Picking a status and double clicking on datagridview activates double click event of datagridview and changes status of product. Second tab pane is for product addition. There are 2 tab panes again. First is for general information, second is for specific. All comboboxes load data from database.

private void loadComboboxStatus()

{

DataTable dt = new DataTable();

string query = "select st\_sales\_description from storemanagement.status\_sales;";

dt = DatabaseFunctions.select(query);

comboBoxStatus.DataSource = dt;

comboBoxStatus.DisplayMember = "st\_sales\_description";

}

private void BtnAdd\_Click(object sender, EventArgs e) //erroru var

{

String name, brand, barcode, supplier,

category, purchasePrice, salesPrice,

quantity, type;

name = textBoxName.Text.Trim().ToUpper();

brand = textBoxBrand.Text.Trim().ToUpper();

barcode = textBoxBarcode.Text.Trim().ToUpper();

supplier = comboBoxSupplier.Text.ToString().Trim().ToUpper();

category = comboBoxCategory.Text.ToString().Trim().ToUpper();

type = comboBoxProductType.Text.ToString().Trim().ToUpper();

purchasePrice = textBoxPurchasePrice.Text.Trim().ToUpper();

salesPrice = textBoxSalesPrice.Text.Trim().ToUpper();

quantity = textBoxQuantity.Text.Trim().ToUpper();

if(String.IsNullOrEmpty(name)|| String.IsNullOrEmpty(brand) || String.IsNullOrEmpty(barcode) ||

String.IsNullOrEmpty(supplier) || String.IsNullOrEmpty(category) || String.IsNullOrEmpty(type) ||

String.IsNullOrEmpty(purchasePrice) || String.IsNullOrEmpty(salesPrice) || String.IsNullOrEmpty(quantity))

{

MessageBox.Show("Please fill all fields");

}

else

{

string supplier\_id, category\_id, type\_id;

supplier\_id = DatabaseFunctions.selectScalar(

String.Format("select pk\_sup\_id from storemanagement.suppliers" +

" where sup\_companyname = '{0}' and end\_date is null;", supplier));

category\_id = DatabaseFunctions.selectScalar(

String.Format("select pk\_category\_id from storemanagement.categories " +

"where category\_name = '{0}';", category));

type\_id = DatabaseFunctions.selectScalar(

String.Format("select pk\_prodtype\_id from storemanagement.product\_type " +

"where prodtype\_name = '{0}';", type));

string query = String.Format("insert into storemanagement.products " +

"(prod\_name, prod\_brand, barcode, prod\_quantity, " +

"fk\_prod\_type, purchase\_price, sales\_price, fk\_sup\_id," +

" fk\_category\_id,fk\_sales\_s\_id ) " +

"values ('{0}','{1}','{2}','{3}'," +

"'{4}','{5}','{6}','{7}','{8}',1) ;",name,brand,barcode,quantity,type\_id,

purchasePrice,salesPrice,supplier\_id,category\_id);

bool result = DatabaseFunctions.runner(query);

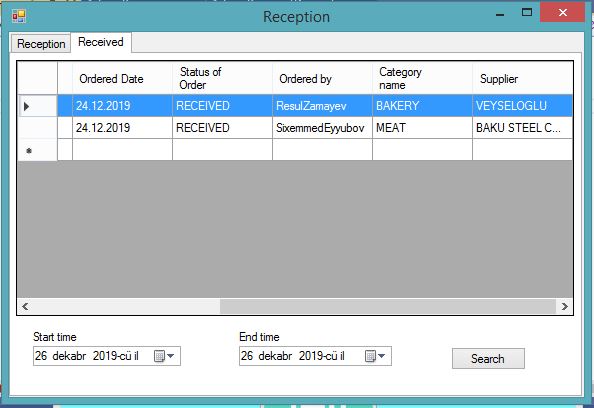
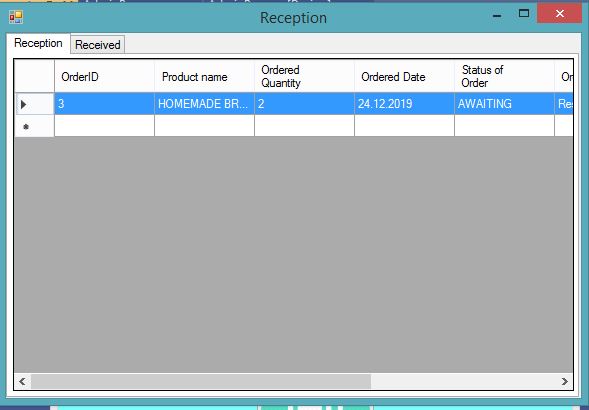
if (result) { MessageBox.Show("Successfully Added!"); }

else { MessageBox.Show("Error in creating new product!"); }

loadElements();

}

}



**Reception page:**

There are 2 pages. First shows ordered products by supermarket. Second shows all received products. In “Reception” double clicking on a line will change status to received.

private void loadReceived()

{

dataGridView1.DataSource = DatabaseFunctions.select("select \* from storemanagement.view\_received ");

}

private void DataGridView2\_DoubleClick(object sender, EventArgs e)

{

string orderid = dataGridView2.SelectedRows[0].Cells[0].Value.ToString();

string update = String.Format("update {0} set fk\_ord\_status\_id = 2 where pk\_order\_id = '{1}' "

,DatabaseFunctions.dbOrders,orderid);

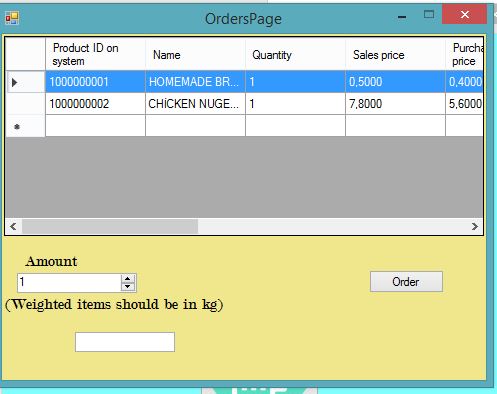
bool res = DatabaseFunctions.runner(update);

if (res) { MessageBox.Show("Received!"); }

else MessageBox.Show("Error in reception");

loadReception();

}



**Orders Page:**

Here a supermarket can make product orders from suppliers. Entering quantity of product to numeric updown and clickin order button is enough.

private void BtnOrder\_Click(object sender, EventArgs e)

{

string orderAmount = numericUpDown1.Value.ToString();

string type = dataGridView1.SelectedRows[0].Cells[6].Value.ToString().Trim().ToLower();

string productID = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

if(type == "weighted")

{

orderAmount = (numericUpDown1.Value \* 1000).ToString();

}

string query = String.Format("insert into {0}(order\_quantity,order\_date," +

"fk\_ord\_user\_id,fk\_ord\_prod\_id,fk\_ord\_status\_id) " +

"values('{1}',date(sysdate()),'{2}','{3}',1);",

DatabaseFunctions.dbOrders, orderAmount, myOrderUserID, productID);

textBox1.Text = query;

bool res = DatabaseFunctions.runner(query);

if (res) { MessageBox.Show("Product ordered!"); }

else { MessageBox.Show("Error in order process"); }

}

**Database functions class:**

Most used functions collected in this class. As a result, all forms can use its functionalities.

public class DatabaseFunctions

{

static int userId;

const string mySQLConnectionString = @"server=localhost;user id=csharp;password=data1234;database=storemanagement";

static MySqlConnection conn;

public const string dbUsers = "storemanagement.users";

public const string dbCategories = "storemanagement.categories",

dbOrders = "storemanagement.orders",

dbProduct\_Type = "storemanagement.Product\_Type",

dbProducts = "storemanagement.Products",

dbReceptions = "storemanagement.Receptions",

dbSales = "storemanagement.Sales",

dbSales\_Detailed = "storemanagement.Sales\_Detailed",

dbStatus\_Orders = "storemanagement.Status\_Orders",

dbStatus\_Sales = "storemanagement.Status\_Sales",

dbSuppliers = "storemanagement.Suppliers";

private static bool connectToDatabase()

{

conn = new MySqlConnection(mySQLConnectionString);

try

{

conn.Open();

return true;

}

catch (Exception)

{

MessageBox.Show("Connection failed");

return false;

}

}

private static bool closeConnectionToDatabase()

{

try

{

conn.Close();

return true;

}

catch

{

MessageBox.Show("Connection couldn't closed");

return false;

}

}

public static DataTable select(string query)

{

if (connectToDatabase())

{

try

{

string command = query;

MySqlCommand cmd = new MySqlCommand(command, conn);

MySqlDataReader reader = cmd.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(reader);

return dt;

}

catch (Exception e)

{

MessageBox.Show(e.ToString());

};

bool ans = closeConnectionToDatabase();

}

return null;

}

public static bool runner(string query)

{

bool result = false;

if (connectToDatabase())

{

try

{

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = query;

cmd.Connection = conn;

cmd.ExecuteNonQuery();

result = true;

}

catch (Exception e) { MessageBox.Show("Runner function error:" + e.ToString()); }

if (!closeConnectionToDatabase()) MessageBox.Show("Connection error Runner");

}

return result;

}

public static int count(string query)

{

int count = -1;

if (connectToDatabase())

{

try

{

string command = query;

MySqlCommand cmd = new MySqlCommand(command, conn);

count = int.Parse(cmd.ExecuteScalar() + "");

}

catch (Exception e)

{

MessageBox.Show(e.ToString());

};

}

return count;

}

public static string selectScalar(string query)

{

string result = null;

if (connectToDatabase())

{

try

{

string command = query;

MySqlCommand cmd = new MySqlCommand(command, conn);

result = cmd.ExecuteScalar().ToString();

}

catch (Exception e)

{

MessageBox.Show(e.ToString());

};

}

return result;

}

public static DataTable getProductsInformation()

{

string query = "select \* from storemanagement.view\_products;";

return DatabaseFunctions.select(query);

}

}

**Result:**

Program successfully runs. It helps supermarket employees to automate business process and keep recordings. It can be used in mini supermarkets. In the project I used my knowledge about SQL queries, joins, software programming concepts. Textbox,combobox,datagridview,tabpane,panel,numeric up-down, label, link label, functions, class logic, events of each used elements, their properities were used for getting prototype.

Here you can get all codes: <https://github.com/Abdurrazakli/UniversityProject/>

**References:**

<https://docs.microsoft.com/en-us/dotnet/framework/winforms/controls/>

<https://www.c-sharpcorner.com/UploadFile/9582c9/insert-update-delete-display-data-in-mysql-using-C-Sharp/>

<http://csharpexamples.com/selectinsertupdatedelete-data-in-mysql-using-c/>

<https://www.codeproject.com/Articles/43438/Connect-C-to-MySQL>